EasyHotspot v0.2 Install Guide

This document version is 0.3 and is intended for EasyHotspot version 0.2 (release) though it should be fine for other versions.

Date: 27th August 2009
Document version: 0.3

This document license is:
Creative Commons Attribution-Share Alike 2.0 UK: England & Wales Licence.
Copyright (c) 2009 Open Mutual Ltd.

Author and Attribution: Lincoln Phipps
Open Mutual Ltd
www.openmutual.net

With special thanks to Rafeequl Rahman Awan who has developed Easyhotspot and all others who have helped patch and test Easyhotspot.

Table of Contents

1 Install Order....4
1.1 Software List....4
1.2 Physical network interface cards....4
1.3 Apache2....5
1.3.1 Chilli - Avoiding use of Self-signed certificates....5
1.4 MySQL....5
1.5 FreeRadius....6
1.5.1 Initial Radius Database setup....6
1.5.2 Easyhotspot Database additions....7
1.5.3 Radius Database users....8
1.5.4 Freeradius Dictionary issues (ChilliSpot attributes)....8
1.5.5 Defining ChilliSpot as a FreeRadius Client....9
1.5.6 Telling FreeRadius to use SQL....9
1.5.7 FreeRadius SQL counter.conf settings needed....11
1.5.8 FreeRadius doesn't start – using Debug mode....12
1.6 ChillSpot - Configuration....13
1.6.1 Chilli - Fixes to allow disconnections from Easyhotspot GUI....13
1.7 Preliminary Client Testing....14
1.8 Easyhotspot Setup....16
1.8.1 Easyhotspot Patches....16
1.8.2 Obtaining and Copying Easyhotspot....16
1.8.3 Mandatory Configuration Settings....17
1.8.4 Other Configuration Settings....18
2 Other Notes....18
2.1 Using Easyhotspot to edit the /etc/chilli.conf file....18
2.2 Language....18
3 Patches....19
3.1 Missing limits in radgroupcheck and radgroupreply....19
3.2 Postpaid Invoice Printing fails due to null stop time....21
3.3 Online Users not listed in Easyhotspot GUI....21
3.3.1 Removing phantom online users (stuck radacct records)....22
3.4 Chilli Disconnect via Easyhotspot GUI....22
3.5 Easyhotspot incorrect editing of /etc/chilli.conf....23
3.6 Vouchers displaying Euro symbol as ? mark....24
3.7 Easyhotspot postpaid decimal handling....24
3.7.1 Decimals - File Changes Overview....25
3.8 Company Tax Code – Configuration option....25
3.9 Typo in Language Parameters - duration_ammount....26
3.10 Typo in Config Parameters - CHILLISPOT_COFIG_FILE....26
3.11 Easyhotspot not setting Acct-Interim-Interval....26
3.12 WISPr bandwidths do not use bits per second values....27
3.13 Radius reply message “Your maximum never usage time has been reached” is confusing. 27
4 Other Useful Software....28
4.1 Monitoring ISP Bandwidth CAPs....28
4.1.1 Installing vnstat for ISP CAP monitoring....28

4.1.2 Installing vnStat PHP frontend....29
4.1.3 Interpreting vnStat results....29
4.1.4 Additional monitoring with bandwidthd....30
4.2 Firewalling....30
5 Notes on using Easyhotspot....30
5.1 Altering postpaid settings....30
5.2 Billing Plans....31
6 TODO....31
6.1 Avoiding persistent MySQL Connections....31
6.2 Account Edit and Add interface feedback quirks....31
6.3 Radacct records persist forever....31
6.4 Taxation handling....31
6.5 Experimental hooks for CoovaChilli....32
7 Document Changes....32
7.1 Version 0.2 preview to 0.2 release upgrade....32
7.2 v0.2 preview to v0.2 release changes....33
7.3 Copying MySQL databases....33

1 Install Order

1. Physical network interface card(s)
2. Install the Operating System. Obviously prefer GNU/Linux distribution
3. Apache2 (with SSL or not)
4. Mysql (if not already working)
5. FreeRadius and default Radius database with Easyhotspot tables
6. Chillispot
7. Initial Client Testing
8. Easyhotspot patches
9. Easyhotspot configuration and acceptance testing.

1.1 Software List

The following is the reference software that I have used. The main issue would be with FreeRadius. There is a dramatic difference between version 1.X and 2.X in the way that FreeRadius organises configuration files. You need ChilliSpot 1.0 to get the ChilliSpot-Max-Total_Octets checks.

`/usr/sbin/freeradius -v`	`FreeRADIUS v2.1.0`
`/usr/sbin/chilli –version`	`Chillispot 1.0`
`/usr/sbin/apache2 -v`	`Apache/2.2.11 (Ubuntu)`
`/usr/sbin/mysqld –version`	`Ver 5.0.75-0ubuntu10.2`
`php –version`	`PHP 5.2.6-3ubuntu4.2 with Suhosin-Patch 0.9.6.2 (cli) and Zend Engine v2.2.0.`
`Easyhotspot`	`0.2 (git) (if you have used 0.2 preview then I show how to upgrade to 0.2 release.`
`My reference OS details`	`Ubuntu Server 9.04 running kernel 2.6.28-15-server`

1.2 Physical network interface cards

You need a machine with a minimum of two ethernet cards. Given how cheap ethernet cards are and how well Linux supports these chipsets then this shouldn't be a problem.

At the end of installing the ChilliSpot then you will also end up with a tunnel, here is an example of a typical machine with the relevant Network Ports,

IP Address	Name	Comments
192.168.182.1	tun0	Created automatically by ChilliSpot
Usually Dynamic	eth0	To Internet router
unnumbered	e.g. eth1	Installed and set according to your machine and as set in the /etc/chilli.conf in the **dhcpif** parameter. Can be ANY physically installed ethernet port.

Note that in my test machine I had 3 ethernet cards, eth0 for the “Office” or administration, eth1 for Internet and eth2 for WIFI so some examples in this document may not use the same numbers as yours. Label the back of the PC in felt-tip with 0, 1 or 2 once you have it working so that when you are recabling you do not make a mistake.

Remember that you can't run another DHCP server on the machine. I am running Ubuntu 9.04 server edition on a Compaq Evo with a P4 1.6 Ghz bought on Ebay (2nd hand) with a 10 Gig hard disk recovered from the trash and 512 Megabytes of memory and 400 Meg of swap. With this machine the typical memory use is 55% and CPU is low so probably could do with more memory if you plan to run a GNOME desktop on the server.

1.3 Apache2

Install this from apt-get or your package manager. We will not cover how to do this and there should be nothing special you need to do to make it work other than that you may not have SSL working due to no openssl and no certificates.

You can either,

- install openssl and generate a self-signed certificate and enable SSL
- or just patch the hotspotlogin.cgi to avoid mandating the need for SSL as described below.

1.3.1 Chilli - Avoiding use of Self-signed certificates

If you want to avoid using a self-signed certificates then you need to do this after you have installed the ChillSpot software. Why you do this is that many browsers will pop-up a very confusing message on the initial connection to your hotspot. If you think it unlikely that anyone can snoop the LAN traffic between the WIFI router and the Easyhotspot machine and you have WPA/WPA2 turned on then you need to do two steps,

a) edit the **/etc/chilli.conf** and alter the **uamhomepage** and the **uamserver** parameters to not refer to https:// style locations.

b) edit the **/cgi-bin/hotspotlogin.cgi** file which checks for if a SSL connection is used to communicate with it and around line 52 comment out that check so that it now says something like,

```
#if (!($ENV(HTTPS) =~ etc etc etc
```

and add a line in the same place that is always false e.g.

```
if (1==0) {
```

which looks stupid in that it will always be false but it means you can easily fix it if you break it by commenting out your line and uncommented the original line.

1.4 MySQL

This must be installed as normal or if it is already installed then you should know the root administrator password or a username with enough privileges to add new databases and users.

WARNING: PLEASE SET A MYSQL ROOT PASSWORD. A new Mysql install will have a blank root password (this is root for MySQL not root for your GNU/Linux)

Open a console and do,

```
mysqladmin -u root password 'new password'
```

e.g.

```
mysqladmin -u root password 'xyzzy'
```

The Easyhotspot database is actually the same as the Radius server database so adding this database will be addressed at a later stage after the Radius is installed. The Easyhotspot needs a database user that is able to update the Radius database as Easyhotspot inserts user and password records into this database. The Easyhotspot database tables are actually a bit of a mess in their naming and should have used table prefixes to allow you to easily pick out the default radius and the Easyhotspot extra tables. Perhaps in a future release we can split these apart.

1.5 FreeRadius

This is installed as normal and it is expected in this document that you'll have a version 2.X of FreeRadius. This version is dramatically different from version 1.X series of FreeRadius in how the configuration files are organised. You need to customise it to specify,

1. a client (which is actually the ChilliSpot software)
2. to use SQL
3. which database
4. defining counters

1.5.1 Initial Radius Database setup

The first thing to do is setup the database used by FreeRadius. Never use the Easyhotspot.sql file that maybe provided to do this as it is a table dump and it will not be fully compatible to the precise Radius software you are using.

Step 1) Create a Radius database. Usually called “radius” but could be called anything e.g. easyhotspot.

In a console do,

```
mysqladmin -u root -ppassword create databasename
```

put in the wanted database where it says databasename and put in the correct password where it says password e.g.,

```
mysqladmin -u root -pxyzzy create radius
```

would create a database called radius.

Step 2) You can either skip to the next section and use the scheme that I have created that has the Radius tables or you can now populate the Radius database with the default Freeradius provided schema. It is recommended that you run this step as your FreeRadius version may have changes to the tables which the Easyhotspot reference database misses,

```
cd /to/path/of/schemas/
mysql -uroot -ppassword database <schema.sql
```

replace password with the right password and the database with the database name e.g.,

```
cd /etc/freeradius/sql/mysql/
mysql -uroot -pxyzzy radius <schema.sql
```

There is a schema.sql file at that path on FreeRadius. You should not need to add the nas.sql or ippool.sql as these are not used. You will now have the following tables,

```
radacct,
radcheck
radgroupcheck
radgroupdreply,
radpostauth,
radreply,
radusergroup
```

These are all MyISAM whereas Easyhotspot is mainly uses InnoDB as it runs transactions. Once that is done then you can now add the contents of the Easyhotspot that are only related to Easyhotspot and the views that it needs.

1.5.2 Easyhotspot Database additions

Easyhotspot doesn't have its own database but uses tables in whatever Radius database you have added plus has a few of its own tables and more importantly it has 3 views that it sets up. We have created a reference sql schema which you can use. The schema names I have created are in 3 files (using "radius" as a database name),

`ehs_withrad_0.2r1.sql`	contains all of the tables including the Radius tables and Easyhotspot tables and views, admin accounts (admin/admin123) and the sample voucher/plan data. Use this if you didn't do the Radius Setup Step 2) in section 1.5.1 above.
`ehs_norad_ 0.2r1.sql`	contains only the Easyhotspot tables and views, admin accounts (admin/admin123) and the sample voucher/plan data (no Radius tables). Use this if you have an existing working Radius database or you did do Radius Setup Step 2) in section 1.5.1 above.
`ehs_norad_nodata_ 0.2r1.sql`	contains only the Easyhotspot tables and views and admin accounts (admin/admin123). Use this if you have an existing working Radius database or you did the Radius Setup Step 2) in section 1.5.1 above but do not want the trial data due to the ehs_norad_0.2r1.sql failing to insert data. You will be able to logon but will have no test data. If you need this then you probably have a different Radius schema.
`ehs_02pre_02rel.sql`	SPECIAL UPGRADE: this contains patches to upgrade an existing 0.2 prerelease to 0.2 release. Use this only if you have already installed a version 0.2 prerelease in the field and want to upgrade to 0.2 release. Do not use if this is a new installation. We do not have a 0.1 to 0.2 upgrade file.

Pick the file you want above and install or update your database with the new file e.g.,

```
mysql -uroot -pxyzzy radius < ehs_norad_ 0.2r1.sql
```

Obviously *xyzzy* is my root password and the *radius* is my suggested database but that could be anything. Delete any residual SQL files from your working area as they may contain passwords. Get copies of these schema from the developer web site.

1.5.3 Radius Database users

There are two users,

- FreeRadius needs to write to the radacct and radpostauth tables
- Easyhotspot needs to write to its own tables plus radcheck, radgroupcheck, radreply, radgroupreply

It is up to you if you want to have one all powerful user or two Mysql user accounts.

FreeRadius server, login, password, radius_db	/etc/freeradius/sql.conf
Easyhotspot database, username and password	system/application/config/database.php

Obviously the FreeRadius radius_db must be the same as Easyhotspot database setting.

Create a file with the following mysql commands substituting the radius@localhost and easyhotspot@localhost and the database name to suite your site,

```
CREATE USER 'radius'@'localhost';
SET PASSWORD FOR 'radius'@'localhost' = PASSWORD('a password');
CREATE USER 'easyhotspot'@'localhost';
SET PASSWORD FOR ''easyhotspot'@'localhost' = PASSWORD('another password');
GRANT SELECT ON radius.* to 'radius'@'localhost';
GRANT SELECT ON radius.* to 'easyhotspot'@'localhost';
GRANT ALL ON  radius.radacct to 'radius'@'localhost';
GRANT ALL ON  radius.radpostauth to 'radius'@'localhost';
GRANT ALL ON radius.* to 'easyhotspot'@'localhost';
```

Run this with mysql on the mysql database (where users are stored) e.g.,

```
mysql -uroot -pxyzzy mysql <accounts.sql
```

where obviously *xyzzy* is my root password and the *mysql* is the specially named mysql database that mysql keeps all its userdata.

1.5.4 Freeradius Dictionary issues (ChilliSpot attributes)

The default dictionary for FreeRadius doesn't know about ChilliSpot attributes that came out in version 1.0 of Chillispot,

- **ChilliSpot-Max-Input-Octets** Maximum number of octets the user is allowed to

transmit. After this limit has been reached the user will be disconnected.

- **ChilliSpot-Max-Output-Octets** Maximum number of octets the user is allowed to receive. After this limit has been reached the user will be disconnected.
- **ChilliSpot-Max-Total-Octets** Maximum number of octets the user is allowed to transfer (sum of octets transmitted and received). After this limit has been reached the user will be disconnected.

The fix is easy. Edit **/etc/freeradius/dictionary** and add in a line,

```
$INCLUDE    /usr/share/freeradius/dictionary.chillispot
```

In Easyhotspot we only use **ChilliSpot-Max-Total-Octets** but note that you will also still need the various radreply, radgroupcheck and radgroupreply related PHP patches in the Easyhotspot to get this into the radgroupcheck and radgroupreply tables to advise FreeRadius about the details of specific billing plans.

If you plan on using Coovachilli then there is **experimental** support for **ChilliSpot-Max-Total-Gigawords** but you must use the special Coovachilli provided dictionary that has this attribute. If you do use Coovachilli because you want to allow sessions larger than 2 Gigabytes then you MUST use the Max-All-MB attributes in your sql counter.conf as ChilliSpot-Max-Total-Octets is limited to integers (signed 32 bits) whereas the Max-All-MB is set as megabytes and so can safely function up to at least 2 Terabytes of traffic.

Coovachilli is experimental because no one has tested this – we have only set the attributes. Make sure that you add php5-gmp support in your PHP if you do this (apt-get install php5-gmp). We need GMP to do maths on arbitrary length integers without truncating them at signed 32 bits. PHP does not support large unsigned integers and certainly not 64 bit unsigned and it is very platform dependant anyway as to what happens.

1.5.5 Defining ChilliSpot as a FreeRadius Client

This is a MANDATORY step. It tells FreeRadius that ChilliSpot is a valid user of FreeRadius. If you do not do this then ChilliSpot cannot use FreeRadius to authenticate.

Edit /etc/freeradius/clients.conf and in the client localhost { section make sure you have,

```
ipaddr            = 127.0.0.1
secret            = <same as radiussecret in /etc/chilli.conf>
nastype           = other
```

1.5.6 Telling FreeRadius to use SQL

We obviously need to use SQL. To do this. Edit **/etc/freeradius/radiusd.conf** and pretty much ignore everything except around about line 640 in the **modules** section uncomment or make sure that you have,

```
$INCLUDE sql.conf
```

and then further down make sure you have,

```
$INCLUDE sql/mysql/counter.conf
```

Then further around line 710 in the **instantiate** section make sure you have,

```
chillispot_max_bytes
noresetcounter
```

or if you plan on going over 2 Gigabytes of user access then,

```
max_all_mb
noresetcounter
```

which are our counters which we define in the next section. Note that you may use max_all_mb instead of chillispot_max_bytes if you want a counter that by implication can go over 2 Gigabytes.

Then in **/etc/freeradius/sites-enabled/default** in the **authorise** section after it has the "Look in an SQL database..." it has an "sql" entry that may be commented out so uncomment it and add the new counters so that it is now,

```
sql
chillispot_max_bytes
noresetcounter
```

or if you plan on going over 2 Gigabytes for your sessions,

```
sql
max_all_mb
noresetcounter
```

and then later in **/etc/freeradius/sites-enabled/default** in the **accounting** section uncomment the sql about line 343,

```
sql
```

and then later in **/etc/freeradius/sites-enabled/default** in the **session** section uncomment the sql about line 370,

```
sql
```

and then finally in **/etc/freeradius/sites-enabled/default** in the **post-auth** section uncomment the sql about line 390,

```
sql
```

That should be it. Now update the **counter.conf** in the next section.

1.5.7 FreeRadius SQL counter.conf settings needed

To match the radcheck and radgroupchecks we use then you also need to add two matching counter.conf checks as follows. Edit the /etc/freeradius/sql/mysql/counter.conf file and in that at the end (unless the counter is already defined in that), add the following,

```
sqlcounter noresetcounter {
counter-name = Session-Timeout
check-name = Session-Timeout
reply-name = Session-Timeout
sqlmod-inst = sql
key = User-Name
reset = never
query = "SELECT SUM(Acctsessiontime) FROM radacct WHERE UserName='%{%k}'"

}

sqlcounter chillispot_max_bytes {
counter-name = ChilliSpot-Max-Total-Octets
check-name = ChilliSpot-Max-Total-Octets
reply-name = ChilliSpot-Max-Total-Octets
sqlmod-inst = sql
key = User-Name
reset = never
query = "SELECT SUM(AcctInputOctets) + SUM(AcctOutputOctets) FROM radacct WHERE
UserName='%{%k}'"

}

sqlcounter max_all_mb {
counter-name = Max-All-MB
check-name = Max-All-MB
reply-name = ChilliSpot-Max-Total-Octets
sqlmod-inst = sql
key = User-Name
reset = never
query = "SELECT SUM(AcctInputOctets)/(1024*1024) + SUM(AcctOutputOctets)/
(1024*1024) FROM radacct WHERE UserName='%{%k}'"

}
```

Make sure that you have set the **radiusd.conf** "instantiate" section and in the /etc/freeradius/sites-enabled/default in the **authorise** section also has this set right as per the previous section to enable FreeRadius to use SQL and to use these counters.

The logic is that initially authorisation is done by SQL then chillispot_max_bytes or the max_all_mb then the noresetcounter in turn. You can add any checks you like here but remember that these checks are only done when the user does a logon and when they are running then these checks are never done again. If you want a check done on a logged in user then it has to be done at the ChilliSpot/Coovachilli which is told what to do in "reply" attributes not the check attributes.

If you want extra checks and need the radcheck or radgroupcheck tables set then please contact the developer. We have added a new config option of access_controller that can be used to switch the logic of the attributes.

1.5.8 FreeRadius doesn't start – using Debug mode

This is pretty quiet when it fails in a module. If you open a browser and do not redirect to the welcome page then the issue is probably part of the FreeRadius configuration. To run FreeRadius in debug mode, from root use,

```
/etc/init.d/freeradius stop
/usr/sbin/freeradius -X
```

and then check the errors or try your logon or whatever you were doing.

1.6 ChillSpot - Configuration

This all works via a single configuration file,

```
/etc/chilli.conf
```

of which you do not need to alter much. The following parameters are the typical ones you change,

radiusserver1	127.0.0.1	Usually unless you have a lot more complicated setup than one machine
radiuserver2	127.0.0.1	Usually unless you have a lot more complicated setup than one machine
radiussecret	Your radius secret	This is the same as **secret** in the /etc/freeradius/clients.conf in the "client localhost {}" section.
dhcpif	ethX	The physical port you want to use that connects to the WIFI router
uamserver	http://192.168.182.1/cgi-bin/hotspotlogin.cgi	Usually this.
uamhomepage	http://192.168.182.1/welcome.html	Usually this.
uamsecret	Same secret that is in the hotspotlogin.cgi	See the hotspotlogin.cgi changes
uamallowed	192.168.182.0/24, 192.168.	Set this to the local tunnel allocation and the DNS server range that the DHCP result has in it.
coaport	3799	UDP port (see RFC 3576) needed to disconnect users.

1.6.1 Chilli - Fixes to allow disconnections from Easyhotspot GUI

By default the chillspot won't disconnect users unless you do a few little tweaks. Edit /etc/init.d/chillispot and look for the DAEMON_ARGS and make sure it is set to,

```
DAEMON_ARGS="--coaport 3799 --conf /etc/chilli.conf"
```

Edit /etc/chilli.conf and add the following line at the end of file,

```
coaport 3799
```

Restart Chillispot with **/etc/init.d/chillispot restart** and then do a **sudo netstat -anp ¬ grep chilli**

to make sure that the two expected port are working i.e. The new 3799 and the original 3990. You will also need to add the Chilli disconnect patch to the Easyhotspot code to correctly call this port from the web GUI. This patch is detailed later in this document.

1.7 Preliminary Client Testing

Before you start testing and installing Easyhotspot then you should have the following netstat results. If you do a **sudo netstat -anp** (as root to get the process names) then you will see many ports in use but you must see these at least,

tcp	0.0.0.0:80	apache2
tcp	0.0.0.0:443	Apache2 (assuming SSL enabled)
tcp	0.0.0.0:3306	mysql
tcp	192.168.182.1:3990	chilli
udp	0.0.0.0:67	dnsmasq
udp	0.0.0.0:1812	freeradius
udp	0.0.0.0:1813	freeradius
udp	0.0.0.0:1814	freeradius
udp	0.0.0.0:3799	Chilli (see note below)

Note on the 3799 port – you must edit the **/etc/init.d/chillispot** script to get this to work. See later details.

The basic flow for a client is,

a) Client PC is setup for DHCP and either connects via WIFI or is plugged into the appropriate port on the Easyhotspot machine
b) The Client PC does a DHCP request (or then DHCL Discover) from 0.0.0.0 to 255.255.255.255 on UDP port 67 as normal
c) This is picked up by the **dnsmasq** process that is part of ChilliSpot but if chillispot has not started then you will never see a DHCP reply if you look on the WIFI ethernet port.

So if you use Wireshark and look on the WIFI ethernet port and see DHCP requests coming into the machine and your firewall is not showing any dropped (using dmesg) and you never see a DHCP reply then check that both the dnsmasq is listening and that the chillispot process is running and listening on port 3990 on the expected IP of 192.168.182.1

netstat -anp | grep 3990 should have
tcp 0 192.168.182.1:3990 0.0.0.0:* LISTEN <pid>/chilli
or something like that.

If all works then the flow is,

From Client		From Chill Server
DHCP Discover	->	DHCP Offer

DHCP Request -> DHCP Ack

or for an existing returning client where it already knows an IP address,

From Client		From Chill Server
DHCP Request	->	DHCP Ack

When the Chilli fails to run then you probably have something wrong with your /etc/chilli.conf file configuration e.g. you have ambitiously uncommented too much !. Note that the Radius need not be correctly working to get the DHCP working. Radius handles the logons not the DHCP and initial redirection web page.

What happens next ?

Assuming that the Client PC has an IP address successful then when you open a browser on the client and try to connect to a web page then the Chilli will reply back with a redirect message of HTTP/1.0 302 Moved Temporarily and provide a location which is the **uamhomepage** setting from the /etc/chilli.conf file e.g. http://192.168.182.1/welcome.html or actually whatever you provided.

You can make that web page as fancy as you like but you **must** provide at least one link to ,

http://192.168.182.1:3990/prelogin

else your clients will never know how to logon ! Do not use https:// in your link to the port 3990 as the ChilliSpot doesn't understand SSL on port 3990 and that will fail with an error message like the connection was interrupted.

If chilli is working on port 3990 then it will redirect you again to the **uamserver** location (from the chilli.conf file e.g. https://192.168.182.1/cgi-bin/hotspotlogin.cgi) but it will provide lots of extra bits in that uamserver URL. If you try to connect to the **uamserver** directly without all the extra bits in the URL that the ChilliSpot has added then it will complain with "Login must be performed through ChilliSpot daemon"

Our basic testing is pretty much finished now and nothing more can be done unless you have installed the Easyhotspot software. Before you do that though you must check as much as possible of the shared secrets between all the parts. You must verify that these are correct manually,

Chilli.conf	**MUST BE SAME AS**
radiussecret	**secret** in the /etc/freeradius/clients.conf in the "client localhost {}" section
uamsecret	**uamsecret** in /cgi-bin/hotspotlogin.cgi

A manual way of testing if the radius is working is to use a shell and enter in,

radtest a a 127.0.0.1 1 *myradiussecret*

and if the myradiussecret is wrong then it will fail telling you that (shared secret is incorrect). The "a" and"a" may not exist. If you had already installed Easyhotspot then there is a default username of "a" and a password of "a". If not then proceed with the Easyhotspot sections below.

Even if you have the **uamsecret** or the **radiussecret** wrong then you will get as far as the ChilliSpot logon screen and will be able to enter your username/password but then after a few seconds chilli tries your username/password but it fails but it won't advise you which is wrong.

If you are at the logon page then you are nearly finished and you can proceed to installing or testing the Easyhotspot software. This is quite separate from the Radius and ChilliSpot in that the Easyhotspot is only ever run to alter the accounts in the Radius database. If you have not got the DHCP working and are not at a ChilliSpot logon page then don't bother starting to install the Easyhotspot. Get ChilliSpot and FreeRadius working first.

1.8 Easyhotspot Setup

Now that you have a working Apache, MySQL, FreeRadius, and ChilliSpot you can safely add the Easyhotspot software. This setup needs doing in this order,

1. Patch the release of Easyhotspot if you wish,
2. Copy or Install to correct location on Web Server,
3. Edit configuration files to suit your personal site,
4. Testing

1.8.1 Easyhotspot Patches

Initial testing was with version 0.2 and though the software looks great a number of issues arose out of that,

- Easyhotspot database unable to be created properly for newer Radius servers,
- Hooks for different counter.conf values and experimental Coovachilli support,
- Checking for NULLs in data,
- Some minor language string errors (in English),
- Missing company business registration (legally mandatory in some countries),
- Bugs in setting maximum session and byte counts,
- Formatting of numbers (maximum decimal places and separators),
- Other minor bugs and enhancements

A list of these patches can be obtained from the developer and you do not really need them to get started but the more important ones are listed later in this document. If you use the current 0.2 release or the preview plus the patches for Easyhotspot then you can be pretty confident that Easyhotspot is very robust.

1.8.2 Obtaining and Copying Easyhotspot

The Easyhotspot version 0.2 preview or above is available from the developers web site at this link http://rafeequl.com/easyhotspot-0.2-preview.zip (the Sourceforge has the old version).

It is a ZIP file so you will need the unzip if you are unzipping on a Linux machine (gunzip won't

do). There is a zip and unzip program available e.g.

```
apt-get install zip
apt-get install unzip
```

In the 0.2 version ZIP file you will find two directory trees, a _MACOSX and the desired easyhotspot-lastest (note the possible spelling mistake).

So download the ZIP file and then do,

```
unzip easyhotspot-0.2-preview.zip
cp -r easyhotspot-lastest /var/www/easyhotspot
```

or whatever path you want to use. It is fine to unzip it straight into your default Web directory at this time e.g./var/www though in the future if Easyhotspot becomes a distro packaged it may reside in /usr/share/ and then you would put in a softlink from your web documents location or use a Apache2 virtual directory. The lazy way is a copy as above. The developer has wrapped the scripts so that they can't be directly accessed so you do not need to protect the directory with a .htaccess file – though you might anyway on a public site or anywhere that people have too much time on their hands.

If you want the latest then to get the latest copy of Easyhotspot from git use,

```
git clone git://easyhotspot.git.sourceforge.net/gitroot/easyhotspot/easyhotspot
```

and then **cp -r** the easyhotspot directory to your /var/www or other htdocs folder.

1.8.3 Mandatory Configuration Settings

You must set the following Easyhotspot configuration settings to get you going with the minimum effort. In version 0.2 we now provide SAMPLE configuration files,

- system/application/config/config.php.SAMPLE
- system/application/config/database.php.SAMPLE
- system/application/config/easyhotspot.php.SAMPLE

On a new system then copy these to their correct names e.g.,

```
cp config.php.SAMPLE config.php
cp database.php.SAMPLE database.php
cp easyhotspot.php.SAMPLE easyhotspot.php
```

and then edit them in turn with your custom changes. On an existing system then compare the .SAMPLE files with your configuration files to see if there are any new parameters you need to set.

In **system/application/config/config.php** and set the base_url to something meaningful. This is where the Easyhotspot administration interface redirects you to no matter how you access it so it can be the same URL as what you use to access your web server with the e.g. http://192.168.5.1/easyhotspot

In **/easyhotspot/system/application/config/database.php** and set the username, password and database to the correct values which you have already defined previously.

If this is done and the MySQL is working then you can connect to your web server and path and you will get the Easyhotspot logon page.

If you have the default administration settings then use the username of admin and a password of admin123 to get you going. Change this once you have logged on.

1.8.4 Other Configuration Settings

Obviously you ideally need to change some other parameters but these are not needed to get you started. These are,

In **system/application/config/easyhotspot.php** set the company_name, company_address, company_address2, company_address3, company_phone, currency_symbol as a minimum.

If you have added the Company Tax Code patch then enter in the **company_tax_code** details as your legal adviser specifies e.g. your VAT , IVA, GST or other tax code that you are legally obliged to add to all invoices.

If you have added the Access Instructions patch then add the **access_instructions** with the suitable text.

If you have added the number format patch then add the **decimal_places**, **decimal_separator** , **thousands_separator**, and **admin_price_input** settings.

If you want to try the experimental Coovachilli support for >2 Gigabytes session bytes then set the **access_controller** to be '**coovachilli**' (and make sure that you have php5-gmp installed). The other options being '**chillispot-hc**' which is what I use and mimics the HotCakes style of counters.conf or the default of '**chillispot-ehs**' which is what you use if you have already defined Max-All-MB counters. Obviously you must install at least CoovaChilli-1.0.13 or higher if you want to use '**coovachilli**' and we don't cover that here but CoovaChilli is a fork of ChilliSpot so there should be very few differences.

2 Other Notes

2.1 Using Easyhotspot to edit the /etc/chilli.conf file

Though I provide a patch to help fix what it does, ideally don't use Easyhotspot but edit the /etc.chilli.conf file manually.

2.2 Language

The language strings are located under...

system/application/language/<language>/easyhotspot_lang.php

e.g. system/application/language/english/easyhotspot_lang.php

The language is set in, **system/application/config/config.php** as the **$config['language']** setting. If you want create a new translation then copy the existing English (ideally) easyhotspot_lang.php to your language director .g.,

system/application/language/**latin**/easyhotspot_lang.php

and set the system/application/config/config.php as the $config['language'] = '**latin**';

Note that not everything is translated and in some cases English is used in the interface but it is nearly all done by the developer.

3 Patches

Only coding patches that cause problems are listed here. The company_tax_code changes I am suggesting affects too many files to easily list so please contact the developer for a updated archive. Easyhotspot will still work without the company_tax_code BUT it won't work exactly right without the following patches I list below.

3.1 Missing limits in radgroupcheck and radgroupreply

The Easyhotspot 0.2 code doesn't add the ChilliSpot-Max-Total-Octets nor Session-Timeout into the radgroupcheck or radgroupreply table. Therefore though you can create an sqlcounter setting with this, it is never verified and neither does ChilliSpot logout an existing logged in user.

The current settings are,

Billing Plan Type	Table	Attribute	Op	Value
Time	radgroupcheck	Max-All-Session	:=	Your limit in seconds
Time	radgroupcheck	Simultaneous-Use	:=	Always 1
Packet	radgroupcheck	Max-All-MB	:=	Your limit in megabytes
Packet	radgroupcheck	Simultaneous-Use	:=	Always 1

The additional settings for all current access_controllers are,

Billing Plan Type	Table	Attribute	Op	Value
Time	radgroupcheck	Session-Timeout	:=	Your session time limit in seconds (used by FreeRadius sqlcounter
Time	radgroupreply	Session-Timeout	:=	Your session time limit in seconds (used by Chillispot

If you are using access_controller of 'chillispot-hc' then you get,

Billing Plan Type	Table	Attribute	Op	Value
Packet	radgroupcheck	ChilliSpot-Max-Total-Octets	:=	Your limit in bytes (used by FreeRadius sqlcounter
Packet	radgroupreply	ChilliSpot-Max-Total-Octets	:=	Your limit in bytes (for the ChilliSpot)

and you do not get the Max-All-MB <<<< this may change to make it more futureproof..

If you are using access_controller of 'coovachilli' then you get,

Billing Plan Type	Table	Attribute	Op	Value
Packet	radgroupreply	ChilliSpot-Max-Total-Gigawords	:=	The upper part of the byte limit when it is over 4 Gigabytes.
Packet	radgroupreply	ChilliSpot-Max-Total-Octets	:=	Your limit in bytes (for the ChilliSpot)

How does this work ?.

1. Easyhotspot inserts the configuration data into the database. It does this independant of FreeRadius or ChilliSpot i.e. You just need MySQL.
2. When you are on a working system then the database is read by FreeRadius whenever there is a logon attempt. FreeRadius looks into the **radcheck** (or **radgroupcheck**) initially based on the sqlcounters (as defined in /sql/mysql/counter.conf) and checks if any of those counters fail.
3. If they do then the user doesn't get to logon at all. If they are all fine then it retrieves the results of the **radreply** or **radgroupreply** and sends those back to the ChilliSpot.
4. FreeRadius is now NEVER used again for authentication checks of any kind but it will process Accounting messages and update radacct with the session details but it never does anything with that. We have used Acct-Interim-Interval (see patch later) to make sure that FreeRadius radacct is updated fairly regularly by ChilliSpot.
5. ChilliSpot understands many radreply or radgroupreply attributes but the most important ones that matter are,
    - ChilliSpot-Max-Total-Octets (and the in and out octets)
    - Acct-Interim-Interval
    - WISPr-Bandwidth-Max-Up and WISPr-Bandwidth-Max-Down
    - Session-Timeout
    - Idle-Timeout
    - (and we also plan to add WISPr-Session-Terminate-Time)

The Coovachilli has also enhanced ChilliSpot and has added ChilliSpot-Max-Total-Gigawords which is the upper 32 bits of a 64 bit counter.
This is why you need to insert the following,

```
ChilliSpot-Max-Total-Octets (and optionally ChilliSpot-Max-Total-Gigawords)
Session-Timeout
```

into **both** the radgroupcheck and radgroupreply else if you only have it in radgroupcheck then the ChilliSpot never stops an existing logged on user. All this is changed in,

```
system/application/models/billingplanmodel.php
```

The edits are too long to detail here. See the developer for new version of the file though you should probably understand the fix that is done. Basically add if() to switch the access_controller config setting and then insert the entries of Max-All-MB, ChilliSpot-Max-Total-Octets, Session-Timeout and Max-All-Session into the correct tables. If someone forks ChilliSpot or Coovachilli again and adds new attributes then we can add a new access_controller and adjust what we set in the tables.

This is all only for voucher customers and is not needed to be done for postpaid customers so we only update the radgroupcheck and radgroupreply not the radcheck or radreply (that the postpaid customers use though a similar logic applies to that too if we add eventually limits to them too).

Please also set the counter.conf correctly in the Install Section of the FreeRadius at the start of this document.

3.2 Postpaid Invoice Printing fails due to null stop time

Edit the system/application/models/invoicemodel.php and around line 68 just before the $this->db->insert add a line,

```
if ($acc_usage['stop'] == null) { $acc_usage['stop'] = $acc_usage['start'] ; }
```

what this does is if an accounting record has been poorly created with a null stop time then the invoice printing fails.

3.3 Online Users not listed in Easyhotspot GUI

There are multiple problems with this in the SQL command. What it is trying to do is rather complicated but the best shortest MySQL command I have found that correctly picks up what it wants to do is ,

```
select username, MAX(acctstarttime) as start, MAX(acctstoptime) as stop,
sum(acctsessiontime) as time, sum(acctoutputoctests +  acctinputoctests) as
packet from radacct group by username having (start > stop) or (stop IS NULL)
```

Why this needs to be done is that a username can have multiple starts and stops and so you will not just get one record per username with a NULL but you may get many records with NULL usually due to your radius server being rebooted. We only want to display the start time of the most recent record but not if it was a record that was disconnected thus the need for the MAX(acctstarttime) and MAX(acctstoptime) and the check to see if the start is greater than the stop. These values that the select produces actually come from different records in the database.

Then the final "having" check is when you have a single entry in the radacct for a logged on user. In

that case there is only one radacct record and it has a start and a stop that is NULL. You can't compare these so we check if the stop is NULL. It never is when you have multiple radacct records.

So that pretty much traps all the weird possibilities that the Radius server can put into the radacct table.

We actually ignore the acctstoptime in the PHP Web GUI (though potentially a view could be created that showed the "Last Logged out time" though the select needs it to filter the results with the "having" command. The aggregate functions on the acctstoptime and packets are self explanitory except to note that the database records with an acctstoptime in them are always zeros.

1) Edit the system/application/models/onlineusermodel.php and look for the db->query(select...) and look at the select and locate,

```
select username, acctstartstime ......
```

2) Copy and paste in the new select command as above.

3.3.1 Removing phantom online users (stuck radacct records)

The Easyhotspot needs a cleanup administration feature that removes stuck accounting records left in the radacct table if there is no voucher for that. Right now the accounting records hang around forever.

This can happen if you change attributes and reboot the server (actually reboot FreeRadius) and do other odd things. You can end up with radacct entries that never go away as "online" users.

With a postpaid account then this is cleared up when they next logon but with a voucher that has expired (time or packets) or you have deleted it then what you must do is very simple.

- Create a postpaid account that has the same name (any password that is easy) as the orphan entry and then logon from a web browser and then disconnect that new session (via logout or manual discconnect).
- Then delete you temporary postpaid account.

The radacct is now cleaned up.

3.4 Chilli Disconnect via Easyhotspot GUI

1) Add the following to the easyhotspot.php config file,

```
$config['radiuscommand'] = "radclient -q -d /etc/freeradius";
$config['radiussecret'] = "your radius secret";
$config['radiusserver'] = "127.0.0.1:3799";
```

2) Edit system/application/controllers/onlineuser.php and make the call in function disconnect()

```
freeradius_disconnectuser($this->uri->segment(3), $this->config-
>item('radiuscommand'),$this->config->item('radiusserver'),$this->config-
```

```
>item('radiussecret'));
```

NOTE to change the segment from 2 to 3 else it will never work.

3) Edit system/application/freeradius_helper.php and make the system call,

```
$result = system('echo "User-Name=\''.$username.'\'" | '. $radiuscommand.' '.
$radiusserver.' disconnect ' .$radiussecret);
```

It is kind of obvious what it is doing as it is sending the shell command. If you are having problems with the user still logged on then try the command from a bash shell without the “q” for quiet switch e.g.

```
      echo "User-Name='yuckt66'" | radclient -d /etc/freeradius 127.0.0.1:3799
disconnect myradsecret
```

(all on one line).

And then also close this file with the php close,

```
      ?>
```

as the developer has not added that. I have got around extra content messing up PHP by passing the “-q” to the radclient command so it is quiet.

If that doesn't work with a short disconnect message then something is not right e.g. the Chilli.conf and chilli have not got the settings right to get the 3799 port working (the coaport).

Note that the radclient MUST have the -d setting as it uses /etc/radiusd or something as a default configuration path and won't find the freeradius location and it MUST have the -q else the system call sends data back to the Web browser that messes with the GUI.

If you want to debug the radcheck then you can remove the -q in the $config['radiuscommand'] and on a disconnect then you will see a disconnect message splatted onto your web browser with result of 41 (if the user was actually connected) or 42 (if the user session was not actually working) and then you will see a PHP error message (which you can ignore).

If you get nothing then your radcheck parameters are broken in some way. Use a shell and identify if it is actually work as expected and that there is a server on the coaport.

3.5 Easyhotspot incorrect editing of /etc/chilli.conf

This patch comes in two parts and as it is quite cumbersome to describe here and unessential I have sent this to the developer. The files you change are,

system/application/libraries/Easyhotspot_chillispot.php

system/application/views/admin/chilli/chilli_view.php

and basically add coaport and fix the breaks in the first file and in the second put in isset() ternary checks to display default data if it is not set.

3.6 Vouchers displaying Euro symbol as ? mark.

This patch is a partial fix for the moment until we can track down the correct setting for dompdf font handling in that I have added a new config setting of currency_symbol_pdf and in system/application/views/voucher/voucher_print.php I have used this instead of currency_symbol.

Set the currency_symbol_pdf to be the currency as plain ASCII text e.g. EURO rather than using a html entity.

TODO: Correctly use the right symbol by setting the font and also allow via change to set the paper size.

3.7 Easyhotspot postpaid decimal handling

This patch has numerous fixes related to invoice calculations. Easyhotspot 0.2 doesn't actually display any decimals at all (uses number_format with hard-coded 0 for decimals). We'll address taxation in a later version of Easyhotspot so right now assume all payments are tax inclusive and there are no pre-tax displayed.

If you need to display pre-tax and taxes paid or have compounding taxes (e.g. Canada) then contact the developer for details and in the meantime you should just consider the “voucher” to be a stock item with a variable price (e.g. like loose fruit) and use your existing till to calculate the tax rates and issue a proper invoice for this sale item rather than relying on Easyhotspot to be a legally compliant billing system. Make each voucher type plus the 2 possible postpaid types different SKUs and set the taxes there.

In addition to the existing currency symbol configuration code there is now the following configuration settings that are used with the PHP number_format() function. I have avoided using locale. Perhaps in the future,

```
decimal_places = <a number> e.g. 2
decimal_separator = ',' or '.' which changes per country convention
thousands_separator = ',', ' ' or .'.'
admin_price_input = 'converted';
```

These are used through out the code. In addition if the admin_price_input is set to “converted” then the prices that are entered by the admin for vouchers are preprocessed by stripping the $_POST['price'] of any thousands_separator and then substituting the decimal_separator with a period (.) symbol used in floating point data. Thus a European pricing input e.g. 1,60 for 1 Euro and 60 cents ends up as 1.60. Note that the postpaid plan doesn't yet have this code – use floating point input.

Where a country uses decimal places in the currency e.g., dollar, Euro or Sterling then you need at least 2 decimal places of resolution when you display your final bill. You can calculate the bill to any decimal place resolution that you like but it should be rounded to the nearest legal money value when it is displayed to the customer according to the usual rules of rounding e.g. bankers rounding or floor or for Australia and New Zealand a method called Swedish Rounding.

At this time we DO NOT ROUND according to a legal convention but the mathematics of truncating floating numbers (which is round half up I think). Its not perfect and a new patch is being worked on to correctly round and handle taxes. It doesn't affect your database, just the displayed

results.

Overall this number_format patch is too large to display here and doesn't affect your Easyhotspot testing. See the developer for more details.

3.7.1 Decimals - File Changes Overview

File	Comment
system/application/views/admin/billingplan/billingplan_view.php	Use all 4 parameters for number_format.
system/application/views/invoices/invoice_detail.php	Use all 4 parameters for number_format.
system/application/views/invoices/invoice_print.php	Use currency_symbol_pdf and all 4 parameters for number_format.
system/application/views/invoices/invoice_view.php	Use all 4 parameters for number_format.
system/application/views/postpaid/postpaid_bill.php	Use all 4 parameters for number_format.
system/application/views/postpaid/postpaid_print_bill.php	Use all 4 parameters for number_format.
system/application/views/postpaid/postpaid_search.php	Use all 4 parameters for number_format.
system/application/views/postpaid/postpaid_view.php	Use all 4 parameters for number_format.
system/application/views/voucher/voucher_print.php	Use currency_symbol_pdf and all 4 parameters for number_format.
system/application/models/billingplanmodel.php	If admin_price_input is set to “converted” then parse $_POST['price'] with thousands_separator and decimal_separator.

3.8 Company Tax Code – Configuration option

I have added a new configuration option of

'company_tax_code' = 'XXXXXXXXX' e.g. the tax registration number.

Which is printed on every invoice. In many countries it is mandatory to display this and it is illegal to issue any invoice or receipt without such a number.

Note that at this time we do not calculate taxes inside Easyhotspot (and truthfully it would make it harder to administer). Please use your existing teller till to do this by creating an appropriate SKU for the types of vouchers or payments.

Overall this patch is too large to display here and doesn't affect your Easyhotspot testing. See the developer for more details.

3.9 Typo in Language Parameters - duration_ammount

The duration_ammount parameter is clearly spelt wrong and should be duration_amount. This is used in,

- easyhotspot_lang.php files for all languages,
- invoice_detail.php
- invoice_print.php
- postpaid_bill.php
- postpaid_print_bill.php

To change you need to set the language files first and then edit each of the other files in turn. It is not service affecting so not a high priority to change.

3.10 Typo in Config Parameters - CHILLISPOT_COFIG_FILE

The CHILLISPOT_COFIG_FILE parameter has that little typo in it should be CHILLISPOT_CONFIG_FILE. This is only used in,

```
system/application/libraries/Easyhotspot_chillispot.php
```

so edit the easyhotspot.php configuration file and that one above.

3.11 Easyhotspot not setting Acct-Interim-Interval

There are two fixes needed here; by default Easyhotspot 0.2 uses a table called “usergroup” whereas by default most Radius installations in their /etc/freeradius/sql.conf will have the usergroup_table as “radusergroup”.

If you do this change then you should delete all prepaid and vouchers plans prior to doing this change as the radacct table is never correctly updated (or manually export/import).

Ideally we should expose the table names to the configuration file but pending that then where “usergroup” appears in,

```
system/application/models/billingplanmodel.php about line 151 and 157
system/application/models/vouchermodel.php about line 25
```

then change this to radusergroup.

The Acct-Interim-Interval is then set in either the radreply for postpaid users or radgroupreply for

vouchers.

The patch is quite large but basically there are now two new configuration parameters,

```
config['postpaid_acct_interim_interval'] = '600';
config['voucher_acct_interim_interval']  = '60';
```

See the developer for the patch.

3.12 WISPr bandwidths do not use bits per second values

There are a few fixes needed. If you add a voucher or billingplan user then the WISPr bandwidth up and down values they get given are not in bps but in kilobits per second and also the 64kbps value is totally wrong at “640”. The value that the database must get given should be in bps.

The code tries to get this right e.g. billingplanmodel.php multiplies the POST value it gets given by 1000 to get a bps value but I think it is safer to keep it as bps everywhere because later the code doesn't divide the database values by 1000. So it is a bit of a mess. If we only ever poke into the database the raw bps values that the WISPr needs then we won't get messed up.

Thus I propose that we only ever display and use bits per second everywhere this bandwidth limit is set.

The values that have to be there have to be e.g. '64000' => '64 kbps' not 64. Edit the,

- system/application/views/admin/billingplan/billingplan_view.php about line 50 and 52,
- system/application/views/postpaid/postpaid_edit.php about line 34 and 38,
- system/application/views/admin/postplan/postplan_view.php about line 31 and 35.

and alter the string as you see fit. I am suggesting the following to the developer,

16000 16 Kbps 32000 32 Kbps 48000 48 Kbps 64000 64 Kbps

96000 96 Kbps 128000 128 Kbps 192000 192 Kbps 256000 256 Kbps

512000 512Kbps 1024000 1 Mbps 2048000 2 Mbps

And so remove the *1000 from the bandwidth calculations at,

- system/application/models/postpaidmodel.php about line 97 and 106
- system/application/models/billingplanmodel.php about line 79 and 88

There are still a few places whereby the value is set but editing the voucher or pre-paid plan doesn't display the current value but this is a low priority to fix.

See the developer for the patch.

3.13 Radius reply message “Your maximum never usage time has been reached” is confusing

This comes back from Radius (rlm_sqlcounter) when you exceed the ChilliSpot-Max-Total-Octets (I guess if you have reply-name = ChilliSpot-Max-Total-Octets in the sqlcounter (sql/mysql/counter.conf) and then try and login with that account. I think you can change this

message in Radius but I couldn't find out where so I did it in hotspotlogin.cgi. Edit hotspotlogin.cgi and around line 371 just after if ($result == 2) { where it prints the failure message then add in,

```
if ($reply =~ /^Your maximum never usage time has been reached$/ ) {
      $reply = "Your credit has expired.";
}
```

or a similar wording as you see fit. If anyone knows how to change this in FreeRadius then please email the developer or me.

4 Other Useful Software

4.1 Monitoring ISP Bandwidth CAPs

Sometimes your ISP may CAP your bandwidth to a contract limit. This certainly is true if you use VSAT (Ku Band or similar Satellite or similar) Internet access in remote locations.

With only one PC you can easily monitor your own personal bandwidth use (on single Windows PC I use Netmeter - http://www.metal-machine.de/readerror/) and you can only blame yourself if you exceed this CAP but if you have customers using your WIFI then you must closely monitor this to make sure that,

- you are correctly pricing your access for your customers to remover your costs,
- are not issuing too many tickets (that is you are not excessively overbooking your bandwidth CAP),
- or if you can better set your bandwidth limits (to use the bandwidth),
- or if you really need to move to a different tariff with your ISP.

To check this I use a program called **vnstat** and there is a separate web interface called **vnStat PHP Frontend**. I also think that Easyhotspot could reuse some of the vnStat PHP (or vnStat SVG) front end code and add this as a feature in Easyhotspot but it is not urgent.

The version I am describing is vnstat 1.6-1 built for Ubuntu and is nicely packaged already plus the vnstat PHP front end version 1.4.1 that is downloaded from www.sqweek.com

4.1.1 Installing vnstat for ISP CAP monitoring

Install vnstat as per normal (apt-get or use your package manager).

On the **first day of use** you need to do,

```
vnstat -u -i ethX
```

where ethX is your ISP facing ethernet port. That's it. If you have other LANs that you want to watch e.g. the WIFI or a private Office LAN then do that for each e.g. on my test system I have three LANs,

```
vnstat -u -i eth0
vnstat -u -i eth1
vnstat -u -i eth2
```

where eth0 is my Office, eth1 is the ISP and eth2 is the WIFI facing.

4.1.2 Installing vnStat PHP frontend

In addition to the vnstat there is a nice front end. This simply calls the vnstat application and displays the result in graphs. It uses GD to draw the graphs.

To install this then download it from the www.sqweek.com web site and unpack it to a suitable location on your web server. e.g. **/var/www/vnstat**

Again you might want to protect this with a **.htaccess** to stop people observing your stats.

Edit the config.php that comes with it,

Find the $iface_list

and set that to a suitable list of ports you want to display **in the order you want them listed** e.g.

```
$iface_list = array('eth1','eth0','eth2');
```

Then locate further down where the $iface_titles are and for each add a $iface_title e.g.,

```
$iface_title['eth0'] ='Office LAN';
$iface_title['eth1'] ='Internet';
$iface_title['eth2'] ='WIFI Physical LAN';
```

4.1.3 Interpreting vnStat results

Vnstat installs a cron file (/etc/crond/d/vnstat) that causes it to run every 5 minutes.
By looking at the various in and out traffic you can identify which part is generating the traffic. The ISP ethernet IN is the sum of the Office and WIFI ethernet OUT and the ISP ethernet OUT is the sum of the Office and WIFI ethernet IN. Any excess in the ISP IN and OUT will usually be the Linux machine itself downloading new packages or remote Internet connections terminating on the Linux machine e.g. SSH connections.

Vnstat will not allow you to identify the type of traffic or IPs – to do that you need to do packet inspection and that is much more complex and only needs to be done if you are getting problems. As far as your ISP is concerned, your traffic is all the same and once you hit the CAP then they either start billing you more or cut your access. Vnstat helps you avoid this problem not identify who to blame.

If you only issue enough vouchers that have packet volumes that add up to your bandwidth CAP (e.g. you issue 100 vouchers per month each with 100 Megabytes thus a 10 Gigabyte total) then you should not have problems unless something has managed to use your WIFI without using a voucher e.g. your personal Office LAN (which vnstat can monitor), someone who is accessing the Easyhotspot machine remotely (without using Radius/Chilli – which you could trace by adding logging to iptables for connections terminating on the Easyhotspot machine).

If you really also need to watch which IP addresses are using traffic then a suitable program is called bandwidthd which I'll also describe.

4.1.4 Additional monitoring with bandwidthd

This comes in two versions, a bandwidthd that doesn't use a database and a Postgresql based version. I'll describe the normal version as few sites would have installed postgresql.

Install the package and then edit the **/etc/bandwidthd/bandwidthd.conf** file and set the subnet to 0.0.0.0/0 and the dev to "ethX" where the X is your ISP facing ethernet port e.g. eth1.

Add a soft link to your web location e.g.

```
cd /var/www
ln -s /var/lib/bandthd/htdocs bandwidthd
```

ideally add a **.htaccess** to that directory to protect it from non-administrators from watching it.

What this now gives you is a more detailed list of the traffic usage on the ISP port. It is not too details but may help you just that little extra without resorting to using Wireshark or EtherApe or some other more sophisticated monitoring tool.

Most people should find that **vnstat** is fine for monitoring ISP Bandwidth CAP on a regular basis.

4.2 Firewalling

This is very complicated and I can't tell you what to use except that you do need to have something. The iptables in Linux is very good but it is quite raw so the way that I handle this is with a program called fwbuilder.

More later on this in a separate document.

5 Notes on using Easyhotspot

To login point your browser at the expected URL e.g. on the server itself you can use 127.0.0.1/easyhotspot or use whatever path you stored Easyhotspot.

You will be met with a logon page. Enter in the default username of admin and the password of admin123

5.1 Altering postpaid settings

When you set this then it applies to all NEW postpaid users that are defined. The fields in the price per MB or price per minute are NOT locale adjusted. Enter them in as floating point.

5.2 Billing Plans

You cannot edit a billing plan. This is used for vouchers. To create a different plan then simply add the new details and then generate vouchers with that plan.

That's all for now.

6 TODO

6.1 Avoiding persistent MySQL Connections

Easyhotspot uses persistent connections (see its database.php config file). If you break Easyhotspot <> MySQL with too many connections then this can jam MySQL from accepting connections from Radius which pretty much makes your server useless. We need to turn this off.

6.2 Account Edit and Add interface feedback quirks

When you are editing cashier or admin/superadmin records then if you enter in data wrong then it puts you back onto the same edit form page but fails to indicate which field has failed its checks and it also throws you at the admin cashier view page.

If you had actually entered in the data correctly then it would have closed the screen fine and correctly updated the database.

So you'll just have to guess which field is bad. You must put in a valid email e.g. x@y.z
not just x@y

You can't also easily delete the admin records. To delete admin records (e.g. if you are cleaning up a database) then use a MySQL editor and remove the record in fa_user and fa_user_profile where the id are the same.

6.3 Radacct records persist forever

The radacct records need two cleanups:

the logic used to see if a username is connected using the radacct record displays records that were in place when the radisus process was killed as always connected even though you have done a disconnect. The fix is to manually touch the acctstoptime (using the acctstarttime).

Plus once the billing has been done for postpaid accounts then the radacct entries for that need to be removed.

6.4 Taxation handling.

Nothing fancy here but ideally a breakdown of pre-tax results. More of this in a separate document.

6.5 Experimental hooks for CoovaChilli

Easyhotspot has some experimental support for CoovaChilli. This is basically the same as ChilliSpot but we will set the ChilliSpot-Max-Total-Gigawords if you define the **access_controller** as '**coovachilli**'.

We use GMP to get arbitrary integer sizes greater than the unreliable platform integer (usually signed 32). You must have the php5-gmp package installed e.g. **apt-get install php5-gmp**

That's all we do. If you find that Coovachilli needs some extra fixes then please contact the developer.

7 Document Changes

This is for the developers to track where this document is up to and to collect stuff which can't go anywhere else.

7.1 Version 0.2 preview to 0.2 release upgrade

Step 1) Backup your mysql database with the command,

```
mysqldump --opt -uroot -ppassword myolddatabase >mysave.sql
```

Step 2) Upgrade the 0.2 preview schema with the command,

```
mysql -uroot -pxyzzy myolddatabase < ehs_02pre_02rel.sql
```

Step 3) Backup your Easyhotspot configuration files with the commands,

```
cd <easyhotspot root>
cd system/application/config
cp easyhotspot.php easyhotspot.02pre.php
cp database.php database.02pre.php
cp config.php config.02pre.php
```

Step 4) Copy in the new release from your unpacked location,

```
cp -r <from easyhotspot v0.2release> <to easyhotspot web root>
```

We are assuming that you have unpacked the tar.gz or zip file to a path that has easyhotspot in it e.g, /home/user/ehs02/easyhotspot and the destination is in e.g. /var/www/easyhotspot thus the command would be,

```
cp -r /home/user/ehs02/easyhotspot  /var/www/easyhotspot
```

Step 5) Go to the Easyhotspot configuration file locations and merge your old configuration back into the new configuration files. The database.php and config.php are easy to do but you may have to manually check the config.php and set new configuration options by hand.

Step 6) You can now login to the new Easyhotspot and test that a client can connect.

7.2 v0.2 preview to v0.2 release changes

This section is developer only. v 0.2 Changes from 0.2 preview (zip) to 0.2 git/release are that there are two new tables of new expiration_account and expirationplan tables and some other schema changes. The differences were found with with mysqldiff (a Perl frontend for MySQL::Diff – see http://adamspiers.org/computing/mysqldiff

Note that you need to do,

```
perl -MCPAN -e 'install Class::MakeMethods'
```

then download the MySQL::Diff tar file and unpack it e.g.,

```
tar xvfz the-tar-gz-file
cd the-mysql-diff-directory
perl Makefile.PL
make
make test
sudo make install
```

then you can use mysqldiff e.g. mysqldiff.pl -n -u root -p password olddatabase newdatabase but don't blindly use that but check it does not have spurious DROP TABLE in it. Note that mysqldiff does display errors for views (I think) so really you need to do,

```
mysqldiff.pl -i -u root -p password olddatabase newdatabase >& mydbdiffs.sql
```

and then manually edit that. It doesn't like views so it throws BOTH sides of the diff at the output. Pick the views that have the "valid_until" in them in the voucher_list. If you do not then you will get a PHP error on the Voucher View talking about Undefined Property of $valid_until. MySQL outputs views using a special set of escape codes that only MySQL understands (prefixing with <!).

TODO: Describe using sed to globally fix database names in difference files.

7.3 Copying MySQL databases

To copy a MySQL database with views you must use a certain option in mysqldump to stop the view getting the old database name e.g.

```
mysqldump --opt -uroot -ppassword myolddatabase >mysave.sql
mysqladmin -uroot -ppassword create newdatabasename
mysql -uroot -ppassword newdatabasename <mysave.sql
```

the mysqldump with "opt" creates a file that can then be used to load any other mysql database. If you find problems with your database then you will need to use mysqldump as shown above to work out what has gone wrong and you can also use this to take backups before you upgrade Easyhotspot.

- end - easyhotspot-v0.2-install-guide-dv0.3.odt